

# J2EE Talking Points

*January 2001*

*Microsoft Confidential—NDA Material for Managed Partners*

On January 16, 2001, Sun held an event for J2EE, inviting press and analysts to come and hear the great things that are happening with their brand. We took the opportunity to come up with some talking points.

Sun's value proposition for EJB and J2EE:

1. offers "faster time to market"
2. eliminates vendor lock-in
3. reduces development costs
4. is open, based on industry-standards

Counterpoints:

- 1. App development today is about interoperability.** Interoperability is about standards. Sun doesn't support standards. They pulled Java out of two standards orgs. They are trailing in all the advanced XML standards. They are not strong participants in groups like the IETF and ISO. Compare their track record with that of Microsoft. We've cooperatively developed SOAP and then submitted it to a standards body. We developed C# and submitted it. We've submitted the CLI. We work with the W3C on important XML standards like Xpath and XML Query, and our tools and runtime are fully compliant on XML standards such as XSLT and Xpath. We're committed to interoperability and the standards that will enable it.
- 2. J2EE is the last of the "big science", big-ticket distributed object projects.** We've repeatedly seen overarching efforts like this fail - look at DCE, CORBA, Smalltalk. None have delivered on their original promise. All have been expensive. IBM, Microsoft, HP, and every other major IT firm are moving away from the "big science" and toward loosely-coupled Web Services. Web Services provide the integration architecture our customers need, not a rip-and-replace, single-language, proprietary solution. Next generation business systems will be built on XML, messaging, and Internet standards. We're building it. We're working with the major players, driving the industry with UDDI. Sun isn't.
- 3. J2EE is fragmented. This recent event is Sun's effort to create a perception of relevancy and consensus behind J2EE.**

Supporting points:

- a. More and more, we are seeing the industry is independently addressing its needs without Sun. For example:
  - SOAP from Apache
  - IBM's Web Services efforts
  - UDDI
  - XML parsers from Apache - (Sun's "specified" parser is not compliant with current standards)
  - Any XML-related technology - XPath, XMLQuery, etc
- b. Why are IBM, Oracle and BEA not only selling their own proprietary extensions, telling customers the core J2EE stuff doesn't work? Example: IBM with JMS; Oracle with Entity Beans; BEA with security model.

- c. This event is all about the J2EE guys screaming to get their money's worth out of this brand. Only BEA and IBM are making any money with Java servers, and they are the vendors who are most independent of the J2EE brand (especially IBM).
- d. The app server market is shaking out, and vendors are moving away from J2EE, not investing in it.
  - Macromedia and Allaire announced a merger.
  - Gemstone swallowed by Brokat; HP swallows Bluestone.
  - Progress Software has totally given up on J2EE.
  - All of the remaining players are struggling, running scared.
- e. Only two companies actually earn revenue from J2EE-related products - BEA and IBM - and both of them are more and more independent of the Sun standard, then what is the meaning or value of J2EE? Even Sun itself (iPlanet) has failed to capitalize on J2EE.

#### **4. In J2EE, Sun is talking about technology that MS has offered for years.**

Supporting points:

- a. MS customers have been doing messaging/component model integration since Nov. '97 (NT Option Pack)
- b. Integrated messaging, component model, transaction services, queued components, etc. -- all part of MS platform
- c. COM+ is the preferred technology versus EJB in terms of real-world development (Evans Data study, Nov 2000)

#### **5. J2EE performance is poor, and the price is high.**

Supporting points:

- a. There are no known, current, public benchmarks that compare the performance of J2EE-based server systems with other offerings. There are no technical reasons why this should be. We believe the Java server vendors are hiding the truth.
- b. The most recent app server benchmark we know of was done in July 1999 by PC Magazine. In it, the Microsoft web application platform, based on Windows NT, bested the performance of the rest of the field by 2:1 or 3:1. All at a system price that was 1/3 to 1/5 the cost of the other guys. Since then, the Java guys have been shy about benchmarks. Is there any wonder?
- c. The world record in the most respected transactional benchmark in the world, the TPC-C benchmark, is held by a Windows 2000 Server system, featuring COM+. In fact, the top 4 marks are all based on COM+ and Windows 2000 Server (as of January 18, 2001). In contrast, there are NO entries for J2EE-based systems.
- d. Two years ago, when business models were scarce, but money was everywhere, many companies bought "big science" systems that required a boatload of expensive consultants to implement them. Now that rationality has returned, some of these companies are in trouble, having spent all their money on overpriced hardware, as opposed to spending it writing software that contributed something unique to their business.

#### **6. Sun is asking users to rip & replace.**

Supporting points:

- a. Enterprise development is a heterogeneous world -- Java is just one of many programming languages that developers know and use.
- b. Customers have massive investments in place that are **not** built on J2EE, and cannot afford the exorbitant cost and excessive time commitment required of replacing and re-educating developers.

- c. With the .NET Framework, Microsoft empowers developers to build and deploy enterprise applications using the language they know best.

**7. Microsoft is actively working with other industry players and standards bodies to establish the truly open, multi-vendor standards that will power the next-gen internet.**

Supporting points:

- a. Work such as our leading role in UDDI with IBM and Ariba, the SOAP and XML Protocols (XP) work with W3C; submitting C# and the Common Language Infrastructure to ECMA, and so on. In contrast, Sun is concerned only with preserving the myth that its technology is somehow an approved standard, while continuing to railroad proprietary technology through its own closed process. Who is open?
- b. Sun reneged on its pledge to submit Java to ECMA and the International Standards Organization (ISO)

## Responding Directly to Sun's Value Proposition

Sun's value proposition in **bold**, with a direct *Microsoft* counterpoint in *italics*:

### 1. **Faster Time to Market - based on server-side components technology**

*Microsoft has been a pioneer in our support of re-usable objects in the form of components. In fact, COM is the most popular component model in the world, with over 200 million deployments. COM has served as the pattern for much Java technology. We agree with this vision, but wonder why Sun and its partners are so late?*

*Sun's server-side component model, EJB, continues to perplex and annoy developers, so that now, more than 2 years after its launch, companies still are not deploying it. In contrast, COM+, delivered in February 2000 with the launch of Windows 2000, is being used and deployed more widely. We have great dot-com success stories featuring customers who went from decision to deployment in under 90 days, some in less than 60 days.*

### 2. **Sun claim: EJB Eliminates Vendor Lock-in**

#### a. **Write-Once Run Anywhere (WORA™)**

*WORA has proven to be an empty promise on the server-side, just as it had previously with client-side java. The fact is that EJB is not a complete application platform. Each EJB vendor has extended the EJB spec in their implementation; any use of these extensions ties a customer into that vendor's product. None but the most simplistic applications will be portable. Even Sun's simple demos are not portable across App Servers.*

#### b. **Large Market of EJB vendors**

*As the space matures, it's clear that only a few vendors will remain as viable options. Giga recently (Aug 2000) commented that consolidation in the EJB-app server space will continue, and that, contrary to Sun's promises, reduced customer choice is the result.*

#### c. **Reduced cost of skills due to standardization**

*Polls repeatedly prove this wrong. In a survey taken in August 2000, in the Philadelphia, Pennsylvania area, Java developers commanded a 14% higher compensation rate, compared to equivalent VB programmers.*

### 3. **Great Connectivity**

#### a. **XML**

*The industry believes in the great potential of XML, and Sun's marketing reflects that. But, in fact, there is no official support for XML in the J2EE spec – no approved or standard APIs, nothing. Sure, people can use XML with Java, but there is no leadership from Sun on this. In contrast, Microsoft has been leading industry efforts on XML for years. Our MSXML parser is rocket-fast, and fully-compliant. Our SQL Server 2000 database delivers XML directly to URL queries. Biztalk Server 2000 consumes and transforms XML. Sun is oddly quiet on the rich topic of XML and what it means for business systems.*

#### b. **JDBC**

*JDBC, Sun's data access strategy, is based on a copy of 3-generations old Microsoft technology called ODBC. JDBC offers poor programmer productivity, and little inherent XML support. Sun's JDO is a technology specification in progress which aims to correct some of these problems, but it is draft stage at this point – nothing usable. In contrast, Microsoft's data access technologies, based on ADO and OLE-DB, with XML support baked in, is miles ahead on programmer productivity and performance.*

c. **JMS, JNDI, JTS**

*These are all minor Java APIs, which are getting lukewarm support from vendors and customers alike. In fact, they are not fulfilling the “great connectivity” promise made by Sun.*